

ОСОБЕННОСТИ АЛГОРИТМА ПОИСКА ОБРАЗУЮЩИХ ПОЛИНОМОВ ДЛЯ ЦИКЛИЧЕСКИХ КОДОВ, ИСПРАВЛЯЮЩИХ ПАКЕТНЫЕ ОШИБКИ

Мышко Е.А.

Научный руководитель: А.Н. Мальчуков
Томский политехнический университет
evgenvt@tpu.ru, lman@tpu.ru

Введение

Для повышения надежности передачи данных применяется помехоустойчивое кодирование [1]. При построении кодового слова циклического помехоустойчивого кода используются образующие полиномы, поиск которых осуществляется с помощью специального алгоритма. Далее будут рассмотрены особенности алгоритма поиска образующих полиномов для циклических кодов, исправляющих пакетные ошибки.

Классификация типов ошибок

Известно большое количество помехоустойчивых кодов, в основу которых положены структурные характеристики кодов. В целом, все коды делят на две самостоятельные группы. К первой относят коды, использующие все возможные комбинации – избыточные коды или в литературе их еще называют простыми. Ко второй группе относятся коды, использующие лишь определенную часть всех возможных комбинаций. Такие коды называются избыточными. Как показал Клод Шенон [2] помехоустойчивость передачи сообщения может быть обеспечена за счет избыточности. Данные коды могут исправлять независимые ошибки и/или пакетные ошибки. На рис. 1, 2 представлены примеры классов ошибок. Независимые ошибки (рис. 1) – это такие ошибки, которые возникают в любом месте кодового слова и их возникновение не зависит от предыдущих или последующих ошибок.



Рис. 1. Независимые ошибки



Рис. 2. Пакетные ошибки

Пакетные ошибки (рис. 2) – это такие ошибки, которые обычно возникают в определённой области кодового слова. Код, ориентированный на исправление таких ошибок, имеет так называемое скользящее окно (рис. 3), все ошибки, попадающие в это окно, исправляются.

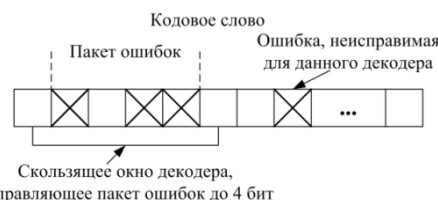


Рис. 3. Пакетные и неисправимые ошибки

Если ошибка не попадает в скользящее окно декодера, то она считается неисправимой для кода (рис. 3).

Особенности алгоритма поиска образующего полинома

Поиск образующего полинома, для циклических кодов, исправляющих независимые и пакетные ошибки осуществляется по двум основным параметрам:

- m – длина информационного блока;
- t – кратность исправляемых ошибок.

Для того, чтобы найти образующий полином необходимо перебрать все полиномы, подходящие под описанные параметры и если среди них не найдётся полином, который пройдёт проверки на необходимые и достаточные условия для образующего полинома, то поиск продолжится среди полиномов, длина которых на единицу больше, чем предыдущих [3]. Необходимое условие:

- строящийся на основе найденного образующего полинома код должен иметь расстояние Хэмминга не меньшее, чем $d = 2*t + 1$ (для независимых ошибок) или $d = t$ (для пакетных ошибок).

Достаточное условие:

- остатки от деления всех комбинаций ошибок на образующий полином в рамках заданной t должны быть уникальными и при этом полином $x^n + 1$ делится на образующий полином без остатка, где $n = m + k$, (k – длина контрольного блока, n – длина кодового слова).

Далее приведено пошаговое описание алгоритма поиска образующего полинома для построения кода, исправляющего независимые ошибки [4]:

Начало.

Шаг 1. Задаются m – разрядность информационного блока и t – желаемая корректирующая способность кода.

Шаг 2. Вычисляется минимальная длина контрольного блока (старшая степень

образующего полинома) по формуле для поиска «границы Хэмминга»[3]:

$$k = \left\lceil \log_2 \left(\sum_{i=0}^t C_n^i \right) \right\rceil$$

где k – старшая степень образующего полинома, t – количество независимых ошибок, исправляемых кодом, n – разрядность кодового слова.

Шаг 3. Выбирается начальное значение веса полинома $w = dm$. Минимальное расстояние кода, исправляющего t независимых ошибок, вычисляется по формуле [5]: $dm = 2t + 1$.

Шаг 4. Выбирается полином из множества, состоящего из полиномов со старшей степенью, равной k и весом, равным w .

Шаг 5. Строится помехоустойчивый полиномиальный код на основе выбранного образующего полинома и генерируются все кодовые слова.

Шаг 6. Вычисляется минимальное кодовое расстояние построенного кода (путём нахождения минимального веса среди всех кодовых слов, за исключением нулевого слова).

Шаг 7. Если вычисленное кодовое расстояние удовлетворяет заданным требованиям, то переходим на шаг 8, иначе переходим на шаг 11.

Шаг 8. Вычисляются синдромы ошибок путем деления всех комбинаций ошибок на образующий полином.

Шаг 9. Если все синдромы ошибок уникальны для построенного кода, то осуществляется переход на шаг 10, иначе переходим на шаг 11.

Шаг 10. Если полином $x^n + 1$ делится на образующий полином без остатка, где n – длина кодового слова, то алгоритм завершает работу, иначе переходим на шаг 11.

Шаг 11. Если проверены все полиномы из множества (множество задано на шаге 4), то переходим на шаг 12, иначе переходим на шаг 4.

Шаг 12. Если вес w равен k , увеличиваем k на единицу и переходим на шаг 4, иначе увеличиваем вес w и переходим на шаг 4.

Конец.

В данном алгоритме на шаге 8 шаблоны ошибок вычисляются путем перебора всех возможных вариантов независимых ошибок кратности t для кода длины n .

Особенности алгоритма поиска полинома для пакетных ошибок

В отличие от алгоритма поиска образующего полинома для кодов, исправляющих независимые ошибки, алгоритм для пакетных ошибок имеет свои особенности. В шаге 2 минимальная длина контрольного блока будет вычисляться по формуле:

$$k = \begin{cases} t = 1, n + 1 \\ t = 2, 2 * n + 1, \\ t > 2, 2 * n + 1 + \sum_{i=1}^{p-2} n * 2^n \end{cases}$$

В шаге 3 минимальное значение веса полинома выбирается равным корректирующей способности кода. Ключевое отличие заключается в формировании шаблонов ошибок, согласно принципу размещения пакетных ошибок в кодовом слове (рис. 4).

Пакет					Кодовое слово				
1	1	1	0	0	0	0	0	...	0

Пакет					Кодовое слово				
0	1	1	1	0	0	0	0	...	0

Пакет					Кодовое слово				
0	0	1	1	1	0	0	0	...	0

Рис. 4. Шаблоны пакетной ошибки

Заключение

В данной работе рассмотрены особенности алгоритма поиска образующего полинома для построения полиномиального помехоустойчивого кода, исправляющего пакетные ошибки. Приведены необходимые и достаточные условия для образующего полинома. Рассмотрен пошагово алгоритм поиска образующего полинома для помехоустойчивого кода, исправляющего независимые ошибки, относительно которого рассмотрены особенности алгоритма поиска для пакетных ошибок. Установлено, что для алгоритма поиска полинома для кода, исправляющего пакетные ошибки основные особенности заключаются в расчете минимальной длины контрольного блока кодового слова, минимального расстояния кода и в принципе формирования шаблонов ошибок.

Список литературы

1. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования: методы, алгоритмы, применение: учебное пособие: пер. с англ. – М.: Техносфера, 2006. – 320 с.
2. Шеннон К. Работы по теории информации и кибернетики. – М.: Изд-во Инстр. лит., 1963. – 832 с.
3. Мальчуков А.Н. Алгоритмическое и программное обеспечение системы для разработки кодеков помехоустойчивых кодов : диссертация кандидата технических наук : 05.13.11 / Мальчуков Андрей Николаевич; [Место защиты: Том. политехн. ун-т]. – Томск, 2008. – 151 с.: ил. РГБ ОД, 61 09-5/472.
4. Mytsko E. A., Malchukov A. N. Adaptation of technology MPI and OpenMP to search for the generators polynomials // 9th International Forum on Strategic Technology (IFOST-2014): Proceedings, Chittagong, October 21-23, 2014. - Chittagong: CUET, 2014 - p. 5-8.
5. Mytsko E. A., Malchukov A. N. Application of parallel computing technology openmp to search for the generator polynomials // Mechanical Engineering, Automation and Control Systems: Proceedings of International Conference, Tomsk, October 16-18, 2014. - Tomsk: TPU Publishing House, 2014 - p. 1-5.